Question Answering with Attentions Ensemble

Zihan Lin [*1] **Teng Zhang** [*2] **Jason Zhu** [*2]

Abstract

We implement three state of the art question answering models: (Seo et al., 2016), which introduces Bi-Directional Attention Flow (BIDAF) network, a multi-stage hierarchical process that uses bidirectional attention flow mechanism to obtain a query-aware context representation without early summarization, (Xiong et al., 2016) that resolves the traditional problem of not recovering from local maximum corresponding to incorrect answers, and R-Net[1], which combines different novel elements (self-matching attention, pointer networks etc.) to obtain state of the art question answering results.

We compare these different models, and gain intuitive understanding of why they work good or not.

1. Introduction

In this project, our group develops a Deep Learning system for the Stanford Question Answering Dataset (SQuAD), which is a large-scale dataset for reading comprehension and question answering created through crowd-sourcing. Given a context paragraph and a question about that paragraph, our model aims to correctly predict the answer of the question, which is assumed to be a continuous sub-span of the context paragraph. Under this setting, the goal of the model is essentially to correctly predict the starting and ending index of the answer in the context.

2. Model Structure

In this project, we first independently implement three existing models with different attention mechanism, which hopefully captures different aspects of the question/context relationships. We then use the ensemble technique on the three models to produce the final answer predictions.

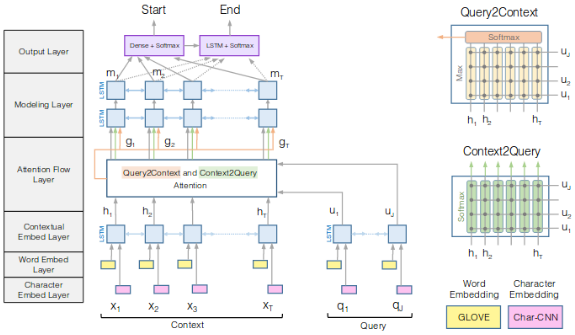

Figure 1. An intuitive understanding of the BiDAF model structure

2.1. BiDAF

The implementation of this model is based on (Seo et al., 2016).

For an intuitive understanding of the model structure, please see figure 4, which is borrowed from (Seo et al., 2016).

Its attention is different from a lot of prior work because:

1. The attention flow layer is not used to summarize the query and context into single feature vectors. Instead, the attention vector at each time step, along with the embeddings from previous layers, are allowed to flow through to the subsequent modeling layer. This reduces the information loss caused by early summarization

2. The attention is computed in two directions: from context to query as well as from query to context. Both of these attentions are derived from a shared similarity matrix

[*] Equal contribution [1] Institution for Computational and Mathematical Engineering, Stanford University, Stanford, USA [2] Management Science and Engineering, Stanford University, Stanford, USA. Correspondence to: Zihan Lin <zihanl@stanford.edu>, Teng Zhang <tengz@stanford.edu>, Jason Zhu <jzhu121@stanford.edu>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

[1] https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf

2.1.1. CHARACTER-LEVEL EMBEDDING

For BiDAF, we also make use of character-level embedding of each word with Convolutional Neural Network (CNN). While we make use of pretrained GLOVE embedding, due to the low dimensional nature of char-level embedding, we train the embedding while we are training the question answering model. The structure of the CNN comes from (Kim, 2014): Characters are embed- ded into vectors, which can be considered as 1D inputs to the CNN, and whose size is the input channel size of the CNN. The outputs of the CNN are max-pooled over the entire width to obtain a fixed-size vector for each word.

Combining the char-level embedding and the word embedding, we pass is to a two-layer Highway Network ((Srivastava et al., 2015)), which is followed by two layers of bi directional LSTMs.

2.1.2. VISUALIZATION OF THE ATTENTION MATRIX FOR AN EXAMPLE

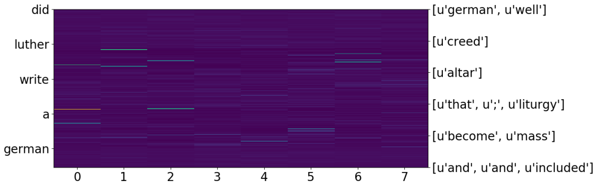

Figure 2. An example: the top attention points for each question word, above a threshold. (The threshold shown in figure is 0.02)

We visualize the attention matrices for some question-context tuples in the dev data in figure 2. The left column 'did luther write a german ?' is the query question, and the right hand side is the words of context that these words pay most attention to.

The second word, luther matches creed and the second word, write matches altar. We find that entities in the question typically attend to the same entities in the context, thus providing a feature for the model to localize possible answers.

2.2. Coattention

The coattention model is based on the work of Xiong et al. (2016). Basically we utilize the two of their contributions. One is introducing the two-way attention structure into our model. The other one is adding the sentinel vectors. The coattention structure is shown in figure 3 from the original paper.

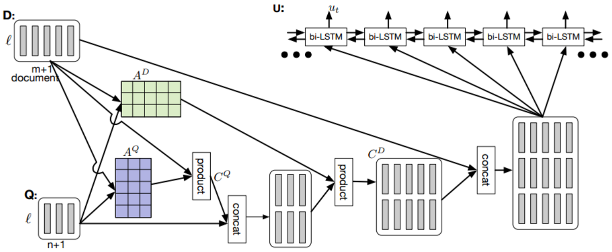

Figure 3. Coattention model structure

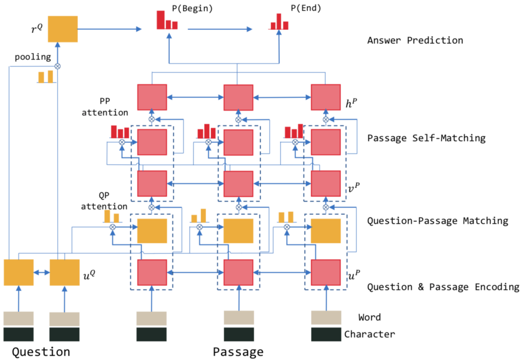

Figure 4. R-Net model structure

2.3. R-Net

The third attentive model we replicated is the R-Net model proposed by Microsoft Research Asia. We followed the structure of the original model, but with some small tweaks. The main structure is as follow:

2.3.1. QUESTION AND PASSAGE ENCODER

For both question $\{w_t^Q\}_{t=1}^m$ and context$\{w_t^P\}_{t=1}^n$, we take their corresponding word-level ($\{e_t^Q\}_{t=1}^m$ and $\{e_t^P\}_{t=1}^n$) and character-level ($\{c_t^Q\}_{t=1}^m$ and $\{c_t^P\}_{t=1}^n$) embeddings, where for the character-level embedding, each character of the word are feed into a bi-directional RNN and the final hidden states are taken for each word. Word-level and character-level embeddings are then concatenated and feed into another bi-directional RNN to produce new representations of the words in both question and context. GRU cells are used for the bi-directional RNN.

2.3.2. GATED ATTENTION-BASED RECURRENT NETWORKS

To incorporate question information into context representation, a question-context matching attention model (Rocktschel et al., 2015) is used and each word of the context is assigned a new hidden state which is a weighted average of the question representations. Finally, an additional gate is used to mask out irrelevant passage parts and emphasize the important ones. To speed up calculation, scaled dot-product attention proposed in (Vaswani et al., 2017) is used:

$$Attention(P, Q) = softmax(\frac{NN(P)NN(Q)^T}{\sqrt{h}})Q$$

where $NN(\cdot)$ represents a dense layer with ReLU activation and dropout.

2.3.3. Self-matching Attention

From the question-aware context representation generated in the last step, we further applied a self-matching attention layer, which has basically the same structure as the previous gated attention step, but the attention is applied to against itself. This is to further evidence from the whole context passage according to the current passage word and question information.

2.3.4. Output Layer

A pointer network (Vinyals et al., 2015) is used to predict the starting and ending index of the answer. Given the representation from the self-attention layer, an additional attention mechanism is used, but the weights are used as a pointer to select the starting position and end position using attention-pooling.

2.4. Ensemble

We first take the output probabilities of the starting and ending index of the answers from each model's output layer, and then take the average of the probability over three models. lastly, argmax of the starting and ending index is taken to generate the final answer p^s and p^e.

$$a_i^s = \frac{1}{3}(a_i^{B,s} + a_i^{C,s} + a_i^{R,s})$$
$$a_i^e = \frac{1}{3}(a_i^{B,e} + a_i^{C,e} + a_i^{R,e})$$
$$p^s = argmax(a_1^s, ..., a_n^s)$$
$$p^e = argmax(a_1^e, ..., a_n^e)$$

where $(a_1^{B,s}, ..., a_n^{B,s})/(a_1^{B,e}, ..., a_n^{B,e})$ are the softmax output probability of the BiDAF model for the starting and ending index. Similarly, we have $(a_1^{C,s}, ..., a_n^{C,s})/(a_1^{C,e}, ..., a_n^{C,e})$, and $(a_1^{R,s}, ..., a_n^{R,s})/(a_1^{R,e}, ..., a_n^{R,e})$ the softmax output probability of the Coattention and R-Net model for the starting and ending index.

Method	performance	
	F1	EM
R-Net (tiny-dev)	0.66	0.59
Coattention (tiny-dev)	0.57	0.48
BiDAF (tiny-dev)	0.65	0.59
BiDAF(Official test)	0.71	0.62
BiDAF(Official dev)	0.71	0.61

Table 1. Performance of three models on the official evaluation set

However, we failed to implement this method because we failed to read different models at the same time. However, we are pretty confident that our ensemble method will work and we will work on this in the future.

3. Experiments & Results

3.1. Implementation Details

3.1.1. BiDAF

Although the main idea was borrowed from (Seo et al., 2016), there are a few key differences in implementation:

1. The word embedding matrix are 2 dimensional for each sentence, compared to 3 dimensional for the original paper. (The original paper further slices the sentence into several pieces and gain a higher dimensional representation.) This may be one of the reasons why our model performs worse than the original paper.

2. The character dictionary is read from the character embedding matrix, but the character embedding matrix is trained with CNN structure mentioned before. This is different from the character embedding implementation in R-Net.

3.1.2. R-Net

The word-level and character-level embeddings we used are both pre-trained GloVe vectors. Due to GPU memory limitations, we only use the first 100 dimension of the GloVe vector based on 840B tokens for word-level embedding by (Pennington et al., 2014), and only use the first 30 dimension of the GloVe character embedding[2]. To speed up calculation, scaled multiplicative attention proposed in (Vaswani et al., 2017) is used. All bi-directional RNN use GRU cells. Most parameters are chosen to match the parameters reported in the original R-Net paper.

3.2. Results

3.2.1. Parameter Tuning

Based on the Coattention model, we explore the model parameter tuning and the results are shown in table 2. As we can see, changing values for the parameters under the Coattention model has minor effect on the model performance. However, truncating the context length into length of 100 can drastically harm the performance, while increasing length up to 600 has no benefit. This can be verified in the dev data set: 19% of the answers emerge after length 100, while 99.6% of the answers are within length 300. Similar behavior is observed in other models.

	value	Step	F1	EM
learning rate	**0.001**	**6.5k**	**0.45**	**0.61**
	0.0001	13k	0.4	0.55
	0.01	blow up		
hidden size	200	6.5k	0.45	0.6
	100	**15k**	**0.45**	**0.61**
	300	7.0k	0.44	0.61
embed size	**100**	**6.5k**	**0.45**	**0.61**
	200	6.5k	0.45	0.61
contex len	600	6.5k	0.45	0.6
	100	4.5k	0.3	0.4
	300	**14k**	**0.45**	**0.61**

Table 2. Model performance of Coattention with different parametric value under different scopes

3.3. Comparison between three models

See figures 7 and 10. The R-Net model that we implemented performed the best on the tiny-dev set. To analyze why it outperformed other models, we drew a picture of Venn diagram of the dev set questions correctly answered by the models BiDAF and R-Net. We also break this comparison down by the first words in the questions.

The answers that get correctly answered by R-Net and not by BiDAF does not have a clear pattern. This may have something to do with the fact that neural architectures are able to exploit much of the information captured by the language feature. By breaking this comparison down, we find that R-Net outperforms the traditional baseline comfortably in every category.

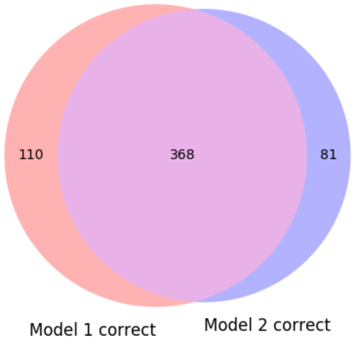

Figure 5. Venn diagram of the questions answered correctly by BiDAF (Model 2) and R-Net (Model 1)

4. Examples of errors and the intuition behind them

Table 3 summarizes some examples where our model made errors, from which we can compare with the true answer. By this, we can get an intuitive understanding of how our model performs and where are the current limitations.

By table 3 and closely looking at the errors our model make, we find that our model does a pretty good job because most of the errors made (about 44% are because of imprecise answer boundaries.) This proves that our question answering models perform better in sense of human understanding than the EM scores tell us.

5. Conclusion

All of our three models obtain superior performance to the original baseline model, and does reasonable performance on most of the errors they make (section 4). By comparing three models, we see the advantages and disadvantages of different models, and gain intuitive understanding of why these models work so well in question answering tasks.

6. Future Work

Ensemble is a quick and naive way to combine multiple models. If we have more time, we can try to combine layers and great ideas from different papers into a single model, which might have better performance and interpretability.

Also, we failed to implement state of the art results claimed in the original papers. This may be due to the changes for the model we make for simplicity, and may be also because

[2] The character-level pre-trained GloVE embedding is downloaded from https://github.com/minimaxir/char-embeddings

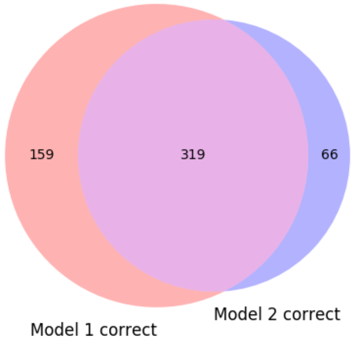

Figure 6. Venn diagram of the questions answered correctly by Coattention (Model 2) and R-Net (Model 1)

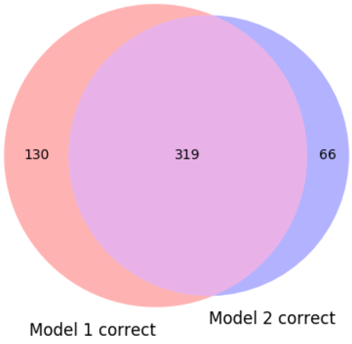

Figure 7. Venn diagram of the questions answered correctly by Coattention (Model 2) and BiDAF (Model 1)

of the fact that we did not have enough time and computing resources to tune different set of parameters.

References

Kim, Yoon. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Rocktschel, Tim, Grefenstette, Edward, Hermann, Karl Moritz, Kocisk, Toms, and Blunsom, Phil. Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664, 2015. URL http://dblp.uni-trier.de/db/journals/corr/corr1509.html#RocktaschelGHKB15.

Seo, Minjoon, Kembhavi, Aniruddha, Farhadi, Ali, and Hajishirzi, Hannaneh. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

Srivastava, Rupesh Kumar, Greff, Klaus, and Schmidhuber, Jürgen. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Ł ukasz, and Polosukhin, Illia. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Vinyals, Oriol, Fortunato, Meire, and Jaitly, Navdeep. Pointer networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2692–2700. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5866-pointer-networks.pdf.

Wang, Wenhui, Yang, Nan, Wei, Furu, Chang, Baobao, and Zhou, Ming. Gated self-matching networks for reading comprehension and question answering. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Xiong, Caiming, Zhong, Victor, and Socher, Richard. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.

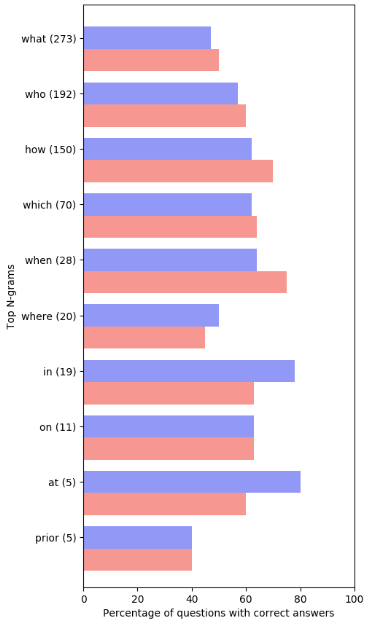

Figure 8. Correctly answered questions broken down by the 10 most frequent first words in the question, BiDAF (Blue bar) and R-Net (Red Bar)

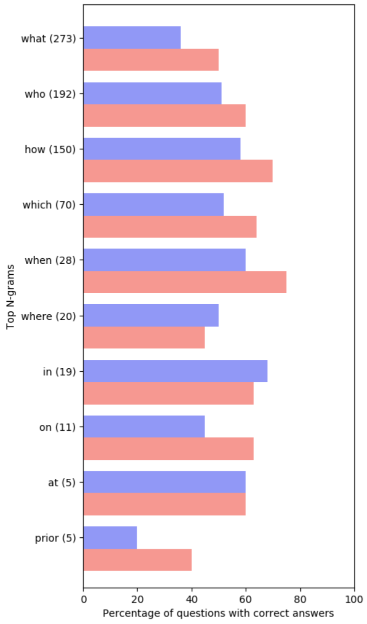

Figure 9. Correctly answered questions broken down by the 10 most frequent first words in the question, Coattention (Blue bar) and R-Net (Red Bar)

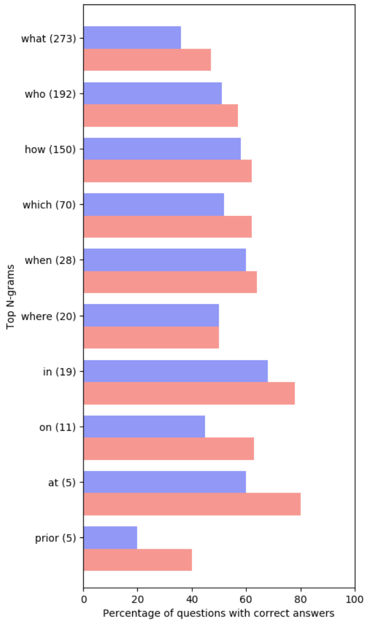

Figure 10. Correctly answered questions broken down by the 10 most frequent first words in the question, Coattention (Blue bar) and BiDAF (Red Bar)

Error type	**Example**
Imprecise answer boundaries	**Context:** ...during that year , tesla worked in pittsburgh , helping to create an alternating current system to power the city 's streetcars ... **Question**: what did tesla work on in 1888 ? **True Answer**: system to power the city 's streetcars **Predicted Answer**: helping to create an alternating current system to power the city 's streetcars
Syntactic complications and ambiguities	**Context:** ...if the head of government of a country were to refuse to enforce a decision of that country 's highest court , it would not be civil disobedience... **Question**: what does not constitute as civil disobedience ? **True Answer**: refuse to enforce a decision **Predicted Answer**: refuse to enforce a decision
External knowledge	**Context:** ...atp synthase uses the energy from the flowing hydrogen ions to phosphorylate adenosine diphosphate into adenosine triphosphate , or atp . because chloroplast atp synthase projects out into the stroma , the atp is synthesized there... **Question**: what does atp synthase change into atp ? **True Answer**: phosphorylate adenosine diphosphate **Predicted Answer**: stroma
Paraphrase problems	**Context:** the plague theory was first significantly challenged by the work of british bacteriologist j. f. d. shrewsbury in 1970 , who noted that the reported rates of mortality in rural areas during the 14th-century pandemic were inconsistent with the modern bubonic plague , leading him to conclude that contemporary accounts were exaggerations .... **Question**: what did shrewsbury note about the plague ? **True Answer**: rates of mortality in rural areas during the 14th-century pandemic were inconsistent with the modern bubonic plague **Predicted Answer**: contemporary accounts were exaggerations

Table 3. A classification of some errors made